Human Control of Air Traffic Trajectory Optimizer

Oliver Turnbull Deimos Space
Email: oliver.turnbull@deimos-space.com
Arthur Richards Department of Aerospace Engineering
University of Bristol, UK
Email: arthur.richards@bristol.ac.uk

***Abstract*—Supervisory constraints are developed for a trajectory optimizer for air traffic. By choosing to apply combinations of these constraints, a human controller can exercise intuitive influence over how conflicts between aircraft are resolved. This offers a compromise between the flexibility of an automated optimizer and the insight of a human controller. Requirements such as the sense of resolution – *i.e.* which aircraft goes first or over – are encoded as constraints on a Mixed-Integer Linear Program (MILP). Examples verify that the constraints work as expected and that the computation times required are reasonable.**[1]

I. Introduction

The Single European Skies [1] and American NextGen [2] programs have both been established in response to increasing passenger numbers and financial and environmental pressures. Both programs aim to address these problems through innovative and more flexible use of the airspace. Central to the future concept of operations [3] is the need to support Air Traffic Controllers (ATCs) with increased levels of automation within the Air Traffic Management (ATM) systems. The current most advanced systems in ATM, such as trajectory and conflict prediction, operate around level one on the ten point scale of automation defined by Parasuraman *et. al.* [4]: "the computer offers no assistance; human must take all decisions and actions".

Trajectory optimization is a performance-driven approach to conflict resolution [5] and has been well studied by ATM researchers [6]–[14]. These works all show that numerical optimization offers good performance, through capturing system objectives in the cost function, with conflict avoidance, achieved by putting separation requirements in the constraints. Optimizing trajectories is also consistent with the idea of the 4-D Reference Business Trajectory at the heart of the SESAR Concept of Operations [3]. However, none of these works consider the ability of a human controller to influence the decision-making process, effectively raising the level of automation straight to four out of ten: "the computer suggests one alternative". Controller workload is considered in [15] by minimizing the number of controller actions required, but still without enabling direct controller input. It would be relatively straightforward to adapt the methods for human supervision via adjustable weights in the objective function. However, such tuning is practically challenging and effectively makes the problem a multi-objective optimization, which are much harder to solve [16].

Instead of adjusting control weights, this paper proposes a solution based on adding constraints to influence the outcome. The approach is inspired by the "playbook" approach, proposed by Miller and Parasuraman [17], [18], in which the supervisor can request certain attributes or actions, but delegate remaining flexibility to the "subordinate" optimization. This enables the controller to influence outcomes in a fast and intuitive manner, without needing to model all of their expertise in the optimizer. For example, a controller may know that wind conditions make the climbing performance of aircraft uncertain, and therefore request that separation is maintained horizontally rather than vertically. We refer to this a restraining the "class" of the resolution to be horizontal. Furthermore, the controller can also influence the "sense" of the resolution, for example which aircraft goes first in a merge, perhaps to account for downstream preferences like arrival sequencing. This enables a more gradual introduction of automation, with the controller and optimizer cooperating to evaluate alternatives, around level three out of ten: "the computer narrows the selection to a few."

This paper adopts a Mixed-Integer Linear Programming (MILP) optimization, common in ATM work [6]–[8], [10], [11], [13]–[15]. MILP captures global decision-making, such as sequencing or choice of sense, through discrete decision variables, and can be solved efficiently despite its complexity using commercial solvers like Gurobi [19]. After a review of the optimization model

[1]This paper describes results from the SUPEROPT project that is part of SESAR Work Package E, which is addressing long-term and innovative research.

in Section II, the controller constraints are introduced in Section III. Example scenarios and computation results are presented in Section IV. Some brief remarks regarding implementation issues are provided in Section V followed by conclusions in Section VI.

II. Model

This section reviews the MILP trajectory optimization framework including aircraft modelling [13] and separation constraints [20]. The aircraft dynamics and kinematics are modelled as a simple point mass approximation:

$$\mathbf{r}(a,k+1) = \mathbf{r}(a,k) + \mathbf{v}(a,k)\Delta t + \frac{1}{2}\mathbf{f}(a,k)\Delta t^2 \quad (1)$$

$$\mathbf{v}(a,k+1) = \mathbf{v}(a,k) + \mathbf{f}(a,k)\Delta t, \quad (2)$$

$\forall a \in \{1,\ldots,N_a\}$, $k \in \{1,\ldots,N_t\}$ where $\mathbf{r}(a,k), \mathbf{v}(a,k)$ and $\mathbf{f}(a,k)$ are the 3D position, velocity and acceleration of aircraft a at time k respectively, and Δt is the time step. The following convex constraints restrict the aircraft dynamics to reflect physical performance limits:

$$\| \left(v_x(a,k), v_y(a,k)\right) \|_2 \leq V_{max} \quad (3)$$

$$\| \left(f_x(a,k), f_y(a,k)\right) \|_2 \leq A_{horiz} \quad (4)$$

$$|f_z(k,a)| \leq A_{vert} \quad (5)$$

$$-D_{max} \leq v_z(a,k) \leq C_{max} \quad (6)$$

$\forall a \in \{1,\ldots,N_a\}$, $k \in \{1,\ldots,N_t\}$ where; V_{max}, A_{horiz}, A_{vert} are the maximum speed, horizontal acceleration magnitude and vertical acceleration magnitude respectively and C_{max} and D_{max} are the maximum rates of climb and descent. In this paper, the maximum rates of climb and descent are constant and not linked to speed. However, the MILP optimization can accommodate more realistic limits using a piecewise affine representation, closely capturing performance models like those found in Eurocontrol's 'Base of Aircraft Data' (BADA) [21]. The reader is directed to [22] for details and examples of this method, which would be especially useful if extending the technique to a terminal area, where separation limits may also be varied. Note that wind is neglected in these examples, but it would be straightforward to introduce a wind estimate into the kinematics equation (1). However, one of the purposes of giving the controller more influence over the optimizer is to account for uncertainty in wind conditions.

The nonlinear two-norm limits in (3) and (4) are inner-approximated as polygons using N_c linear constraints. Hence the speed limit (3) becomes:

$$\mathbf{e}_i^T \mathbf{v}(a,k) \leq V_{max} \cos\left(\frac{\pi}{N_c}\right) \quad (7)$$

$\forall i \in \{1,\ldots,N_c\}$ where $\mathbf{e}_i$ is a horizontal unit vector

$$\mathbf{e}_i = \begin{pmatrix} \cos\left(2\pi i/N_c\right) \\ \sin\left(2\pi i/N_c\right) \\ 0 \end{pmatrix}.$$

Similar constraints are applied for acceleration (4). A minimum speed constraint is also applied

$$\| \left(v_x(a,k), v_y(a,k)\right) \|_2 \geq V_{min} \quad (8)$$

and since this is non-convex, it is implemented using binary decision variables

$$\mathbf{e}_i^T \mathbf{v}(a,k) \geq V_{min} - \left(V_{max} + V_{min}\right) d_v(a,k,i) \quad (9a)$$

$\forall i \in \{1,\ldots,N_c\}$ where $d_v(a,k,i) \in \{0,1\}$ is used to selectively relax its associated velocity constraint, and the following logical constraint ensures that at least one constraint is *not* relaxed:

$$\sum_{i=1}^{N_c} d_v(a,k,i) \leq N_c - 1. \quad (9b)$$

Every aircraft has a reference trajectory including initial plans for position $\mathbf{r}_{ref}(a,k)$ and velocity $\mathbf{v}_{ref}(a,k)$. The initial state of each aircraft is constrained to match that of the reference:

$$\mathbf{r}(a,0) = \mathbf{r}_{ref}(a,0) \quad (10)$$

$$\mathbf{v}(a,0) = \mathbf{v}_{ref}(a,0) \;\; \forall\, a \in \{1,\ldots,N_a\} \quad (11)$$

Also, every aircraft has a target position $\mathbf{r}_{targ}(a)$ and velocity $\mathbf{v}_{targ}(a)$. Although the total number of time steps is limited to a maximum of N_t for every aircraft, individual arrival times are variable, and again encoded using binary variables [23]. The target tolerance $\Delta\mathbf{r}$ is measured at the chosen arrival time

$$\|\mathbf{r}(a,k) - \mathbf{r}_{targ}(a)\|_\infty \leq \Delta\mathbf{r}(a) + M\left(1 - d_a(a,k)\right) \quad (12a)$$

$$\|\mathbf{v}(a,k) - \mathbf{v}_{targ}(a)\|_\infty \leq \Delta\mathbf{r}(a) + M\left(1 - d_a(a,k)\right) \quad (12b)$$

$\forall a \in \{1,\ldots,N_a\}$, $k \in \{1,\ldots,N_t\}$ where $d_a(a,k) = 1$ if and only if aircraft a chooses step k as its finishing time. The tolerance is subject to limits of up to one half the distance traveled in a time step

$$\| \left(\Delta r_x(a), \Delta r_y(a)\right) \|_\infty \leq \frac{1}{2} V_{max} \Delta t \quad (12c)$$

$$|\Delta r_z(a)| \leq \frac{1}{2} D_{max} \Delta t \quad (12d)$$

$\forall a \in \{1,\ldots,N_a\}$. A logical constraint is included to ensure that an arrival time is chosen:

$$\sum_{k=1}^{N_t} d_a(a,k) = 1 \quad (12e)$$

The cost function is the accummulated flight time for all aircraft, plus secondary terms penalizing altitude changes and accelerations:

$$J = \sum_{a=1}^{N_a} \Bigg[\mathbf{w}_f \Delta \mathbf{r}(a) + \sum_{k=1}^{N_t} k d_a(a,k) + w_c |v_z(a,k)| + w_a \|\mathbf{f}(a,k)\| \Bigg] \tag{13}$$

where the weights $\mathbf{w}_f = (1/V_{nom}, 1/V_{nom}, 1/C_{max})$ convert the tolerance into an estimate of remaining travel time. The other weights (w_c, w_a) allow the relative importance of the penalty terms to be tuned.

Separation is defined as a cylindrical protection region around each aircraft:

$$\left\| \begin{pmatrix} r_x(a,k) - r_x(b,k) \\ r_y(a,k) - r_y(b,k) \end{pmatrix} \right\|_2 \geq S_{horiz} \tag{14a}$$

$$\text{OR } |\,(r_z(a,k) - r_z(b,k))\,| \geq S_{vert} \tag{14b}$$

applied for pairs of aircraft (a,b) with $b > a$ and for all time $k \in \{1,\ldots,N_t\}$ where S_{horiz} and S_{vert} are the horizontal and vertical separation requirements between two aircraft. Like the minimum speed constraints (9), the cylindrical exclusion is approximated as a polytope. Binary variables $d_s(a,b,k,i)$ selectively relax each facet and a logical constraint ensures at least one is not relaxed:

$$\mathbf{e}_i^T (\mathbf{r}(a,k) - \mathbf{r}(b,k)) \geq S_{horiz} - M(1 - d_s(a,b,k,i)) \quad \forall i \in \{1,\ldots,N_c\} \tag{15a}$$

$$r_z(a,k) - r_z(b,k) \leq -S_{vert} + M(1 - d_s(a,b,k,N_c+1)) \tag{15b}$$

$$r_z(a,k) - r_z(b,k) \geq S_{vert} - M(1 - d_s(a,b,k,N_c+2)) \tag{15c}$$

$$\sum_{i=1}^{N_c+2} d_s(a,b,k,i) \geq 1 - \sum_{j=1}^{k-1} (d_a(a,j) + d_a(b,j)) \tag{15d}$$

$\forall (a,b) \in \{\{1,\ldots,N_a\} \times \{1,\ldots,N_a\} | b > a\}$ $\forall k \in \{1,\ldots,N_t\}$. Note that (15d) relaxes the separation constraints as soon as one or other aircraft has reached its target.

Furthermore, to enable planning with large time-steps, the method of [20] is adopted such that the same binary variables are also enforced at the previous time-step to ensure that the entire connecting *line-segment* remains conflict free:

$$\mathbf{e}_i^T (\mathbf{r}(a,k-1) - \mathbf{r}(b,k-1)) \geq S_{horiz} - M(1 - d_s(a,b,k,i)) \quad \forall i \in \{1,\ldots,N_c\} \tag{16a}$$

$$r_z(a,k-1) - r_z(b,k-1) \leq -S_{vert} + M(1 - d_s(a,b,k,N_c+1)) \tag{16b}$$

$$r_z(a,k-1) - r_z(b,k-1) \geq S_{vert} - M(1 - d_s(a,b,k,N_c+2)) \tag{16c}$$

III. Supervisory Constraints

This section introduces additional constraints, designed to be selectively introduced by a human controller to influence the solution. The constraints available are summarized in Table I. Each can be introduced for chosen aircraft pairs (a,b) or some for individual aircraft (a). The constraints can be combined, with different choices for different pairs and/or multiple choices for the same pairs. Examples in Section IV will further illustrate their effect. The remainder of this section describes how they are implemented.

TABLE I
Summary of Supervisory Constraints

Sense constraints	
OVER(a,b)	a to pass over b if resolving vertically
BEFORE(a,b)	a to pass ahead of or go in front of b if resolving horizontally
Class constraints	
SEPHORIZ(a,b)	a and b to be always separated horizontally
PATHSEP(a,b)	a and b to be separated spatially, *i.e.* paths must not overlap
FIXHPATH(a)	a to remain on reference 2D path, but speed and height changes permitted
FIXPATH(a)	a to remain on 3D reference path, but speed changes permitted

A. Constraint OVER(a,b)

The first constraint aims to force aircraft a to pass over aircraft b, instead of under it. This sense constraint would be used when a controller feels that the opposite resolution would be problematic, perhaps due to winds affecting climb slopes relative to ground. It is tempting simply to constrain that one aircraft was always above the other, $r_z(a,k) > r_z(b,k)$ $\forall k$, but this is unworkable. For example, if aircraft a starts below b, this instantly renders the problem infeasible. Instead, we constrain only that a can never be *directly below* b. This does not, in itself, prevent other classes of resolution, such as a going behind b, but it can be combined with other tactics

such as PATHSEP to achieve the desired effect. Mathematically, the effect is realized by constraining the binary variable that switches the "a below b" constraint (15c) or, equivalently, the "b above a" constraint (15c).

$$0 = \begin{cases} d_s(a,b,k,N_c+1) & a > b \\ d_s(b,a,k,N_c+2) & a < b \end{cases} \tag{17}$$

B. Constraint BEFORE(a,b)

Like the OVER constraint, this attempts to force a particular sense of resolution, but now in the horizontal case. In 2D, sense selection means specifying aircraft a to go either ahead of, or behind, aircraft b. This constraint would be useful where ground speed uncertainty made the opposite sense too risky, in the view of the controller, or perhaps where a particular sequence downstream was desired. In the vertical case, constraining sense is easier, because all aircraft share a common global Z-axis. However, sense is less clear in the horizontal case, because the global direction of "ahead" or "behind" differs between aircraft. Therefore the sense constraint is applied temporally rather than spatially: aircraft b is prevented from intersecting with any *future* location of aircraft a. This is achieved with additional constraints similar to (15a)–(15d). These avoidance are applied for all pairs of time steps $(j,k) \in \{\{1,\ldots,Nt\} \times \{1,\ldots,Nt\} | j > k\}$:

$$\begin{aligned} \mathbf{e}_i^T(\mathbf{r}(a,j) - \mathbf{r}(b,k)) &\geq S_{horiz} \\ &\quad - M(1 - d_t(a,b,j,k,i)) \\ &\quad \forall i \in \{1,\ldots,N_c\} \end{aligned} \tag{18a}$$

$$\begin{aligned} r_z(a,j) - r_z(b,k) &\leq -S_{vert} \\ &\quad + M(1 - d_t(a,b,j,k,N_c+1)) \end{aligned} \tag{18b}$$

$$\begin{aligned} r_z(a,j) - r_z(b,k) &\geq S_{vert} \\ &\quad - M(1 - d_t(a,b,j,k,N_c+2)) \end{aligned} \tag{18c}$$

$$\sum_{i=1}^{N_c+2} d_t(a,b,j,k,i) \geq 1 - \sum_{n=1}^{j-1} d_a(a,n) - \sum_{m=1}^{k-1} d_a(b,m) \tag{18d}$$

and again, the "previous step" constraints are used for inter-sample separation:

$$\begin{aligned} \mathbf{e}_i^T(\mathbf{r}(a,j-1) - \mathbf{r}(b,k-1)) &\geq S_{horiz} \\ &\quad - M(1 - d_t(a,b,j,k,i)) \end{aligned} \tag{19a}$$

$$\begin{aligned} r_z(a,j-1) - r_z(b,k-1) &\leq -S_{vert} \\ &\quad + M(1 - d_t(a,b,j,k,N_c+1)) \end{aligned} \tag{19b}$$

$$\begin{aligned} r_z(a,j-1) - r_z(b,k-1) &\geq S_{vert} \\ &\quad - M(1 - d_t(a,b,j,k,N_c+2)) \end{aligned} \tag{19c}$$

If the OVER() constraint is used in conjunction with BEFORE(), the relevant d_t binaries in (18b)–(19c) are also constrained to zero.

C. Constraint SEPHORIZ(a,b)

This constraint requires horizontal separation, meaning that any conflict between a and b must be resolved by horizontal separation. Unlike the previous two sense constraints, class constraint are symmetric, so SEPHORIZ(a,b) has the same effect as SEPHORIZ(b,a). It is achieved by combining both OVER(a,b) and OVER(b,a). Having removed the possibility of a passing either over or under b, all that remains is horizontal.

D. Constraint PATHSEP(a,b)

This tactic requires separating the paths spatially, i.e. such that they do not insersect. This is in contrast to separating temporally, such that paths intersect but the aircraft reach the crossing point at different times. PATHSEP is achieved by combining BEFORE(a,b) and BEFORE(b,a). This implies that a is separated from all past and future positions of b. Note the similarity with the double application of OVER() to achieve SEPHORIZ(): essentially, by removing both options for resolving in a particular way (e.g. vertically or temporally), all that remains is to adopt another resolution (e.g. horizontal or spatial, respectively).

E. Constraints FIXPATH(a) and FIXHPATH(a)

These two constraints apply to only one aircraft, fixing its behaviour relative to the reference trajectory $\mathbf{r}_{ref}$. FIXPATH constrains the new trajectory in 3D (but crucially not 4D): the aircraft must follow the same path as before, but not necessarily at the same speed. FIXHPATH fixes only the horizontal path, such that height and speed changes can be effected. These constraints could be used to limit ATM commands to only speed or height advisories, where conditions would make lateral (or vertical, in the case of FIXPATH) resolutions undesirable. Because the timing at each 3D (or 2D) point can change, this is not as simple as fixing the decision variables, which are each tied to a time. Therefore, a piecewise

affine formulation [24], [25] is applied:

$$\forall k = \{1, \ldots, N_t\} :$$

$$\mathbf{r}(a,k) - \sum_{j \in \mathcal{J}(k)} \lambda(a,k,j)\mathbf{r}_{ref}(a,j) \in \mathcal{D} \quad (20a)$$

$$\lambda(a,k,j) \in [0,1] \quad (20b)$$

$$\sum_{j \in \mathcal{J}(k)} \lambda(a,k,j) = 1 \quad (20c)$$

$$d_p(a,k,j) + d_p(a,k,j+1) \geq \lambda(a,k,j) \quad (20d)$$

$$\sum_{j \in \mathcal{J}(k)} d_p(a,k,j) = 1 \quad (20e)$$

$$d_p(a,k,j) \in \{0,1\} \quad (20f)$$

where $\mathcal{D}$ represents the constraint on deviation from the reference path. For the FIXHPATH tactic, $\mathcal{D}$ permits only a small horizontal deviation but does not limit vertical change. For the FIXPATH tactic, there are also limits on the Z-axis difference. The binary variable $d_p(a,k,j) = 1$ if and only if the new point k of the trajectory is to be placed on the segment from j to $j+1$ of the reference trajectory. The decision variables λ are then constrained such that the new point is expressed as a weighted combination of the endpoints of the chosen segment. The index set $\mathcal{J}(k) := \{j \in 1, \ldots, N_t \mid |j-k| \leq 2\}$ permits each new chosen point to "slide" along the reference path by up to two time steps. This limits the number of binary variables d_p introduced.

IV. Examples

This section illustrates the use of the tactic constraints through examples of air traffic scenarios. The settings for all scenarios are summarized in Table II. The units are unusual – nautical miles; kilofeet; and minutes – but enable a reasonable air traffic scenario to be handled with simple scaling and comparable orders of magnitude on the decision variables. The optimizations were modeled in AMPL and solved with Gurobi 6.0.4 on a PC with an Intel Core i7-3770S 3.10 GHz quad core processor and 8GB RAM running Windows 7. Matlab was used to generate the example data and visualize the results.

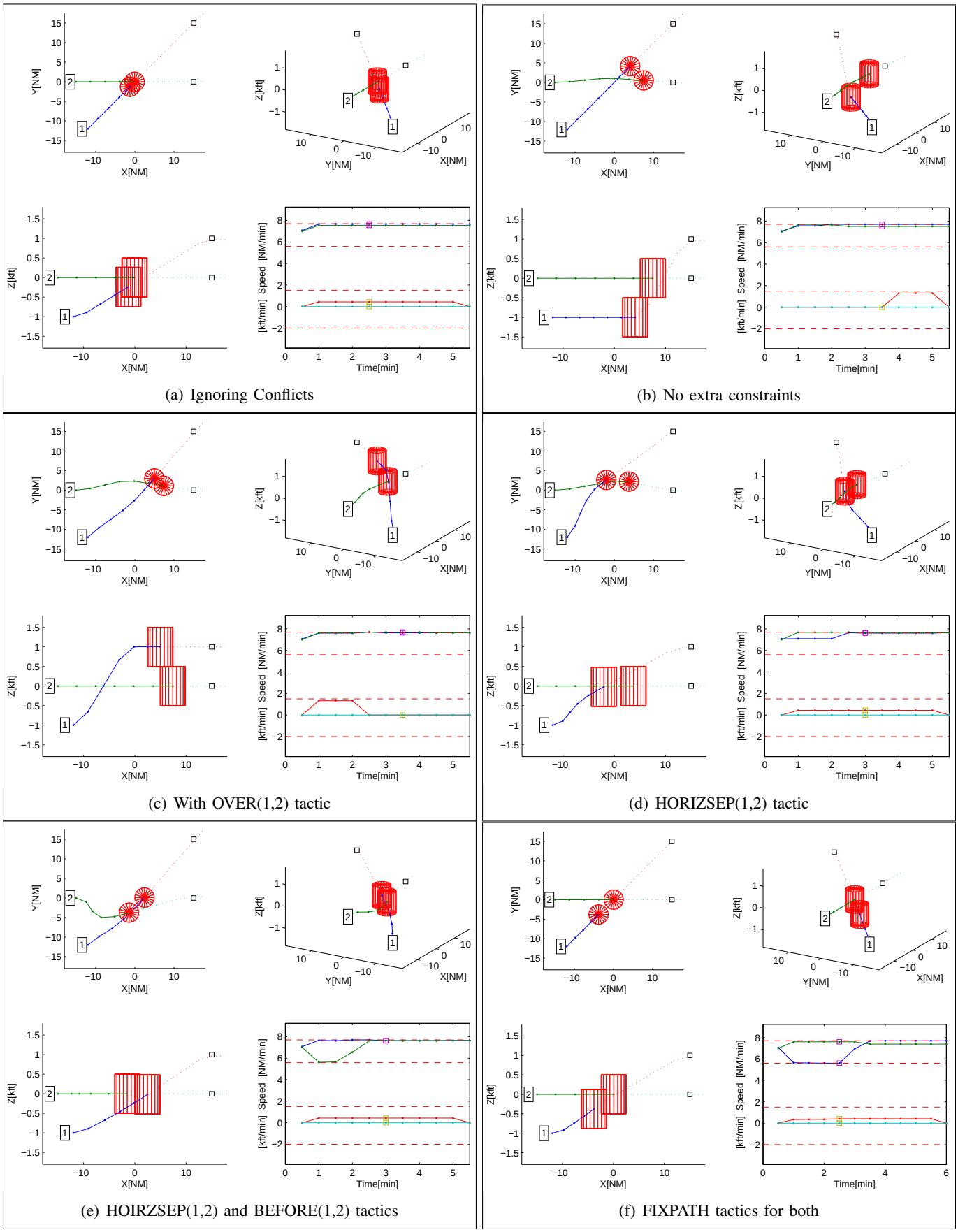
Figure 1 shows one scenario solved with many different constraints. In all cases, Flight 1 is climbing from 1000 ft below Flight 2 to 1000 ft above it, while crossing its path. For each solution, the figure shows a plan view (X-Y), a side view (X-Z), an isometric view (to aid visualization of the separation cylinders in 3D), and speed histories. The latter include two lines for each aircraft: one in the range -1.5 to 2 showing the vertical speed in kft/min and the other between 5.6 and 7.7 showing the horizontal speed in NM/min. The relevant limits are shown dashed. Each plot also highlights a particular instant in the solution trajectories with cylinders of radius 2.5 NM and height 0.5 kft, such that separation is achieved if no two cylinders overlap. The numbers identify the starting point of each flight and the squares mark the targets.

TABLE II
Settings for example scenarios

Quantity	Value	Comment
Horizontal space units	NM	Nautical miles
Vertical space units	kft	1000s of feet
Time unit	min	Minute
Nominal speed	7.0 NM/min	420 knots
Maximum speed V_{max}	7.7 NM/min	Nominal +10%
Minimum speed V_{min}	5.6 NM/min	Nominal -20%
Maximum climb rate C_{max}	1.5 kft/min	Estimates from BADA for flights at around 30,000ft
Maximum descent rate D_{max}	2 kft/min	
Maximum horizontal acceleration A_{horiz}	11 NM/min^2	Permits a 1.5^o/s turn at nominal speed
Maximum vertical acceleration A_{vert}	58 kft/min^2	Equivalent to $0.5g$
Horizontal separation S_{horiz}	5 NM	Typical en-route separation
Vertical separation S_{vert}	1 kft	1000ft reduced vertical separation minimum
Climb cost weight w_c	1 min/kft	
Acceleration cost weight w_a	10^{-5} min/unit	
Sector size	30 x 30 NM	
Time step Δt	0.5 min	For limited bandwith
Max. number of time steps N_t	12	Enough for 42NM flight at nominal speed

Figure 1(a) shows the solution ignoring conflicts, and shows that there is indeed a loss of separation. Figure 1(b) shows the default answer when separation constraints are applied but without any tactics. In this case Flight 1 stays low until it has passed below Flight 2, before climbing to its destination. When the OVER(1,2) constraint is applied in Figure 1(c), Flight 1 climbs earlier to pass over Flight 2, as requested. Note that Flight 2 also diverts slightly to give more room for 1 to climb. In Figure 1(d), the HORIZSEP tactic prevents passing over or under, so Flight 1 performs a steady climb while both divert left, such that Flight 1 passes behind 2. Figure 1(e) further constrains the case of horizontal resolution to be the opposite from the default in Figure 1(d), resulting in a sharp right turn by aircraft 2 to pass behind the back of aircraft 1. Finally Figure 1(f) shows the results when the FIXPATH constraint is applied to both aircraft, meaning that only a speed resolution is available, and aircraft 1 is seen to reduce its speed to the minimum, so as to let aircraft 2 pass ahead.

Figure 2 shows a scenario in which two aircraft at the



(a) Ignoring Conflicts

(b) No extra constraints

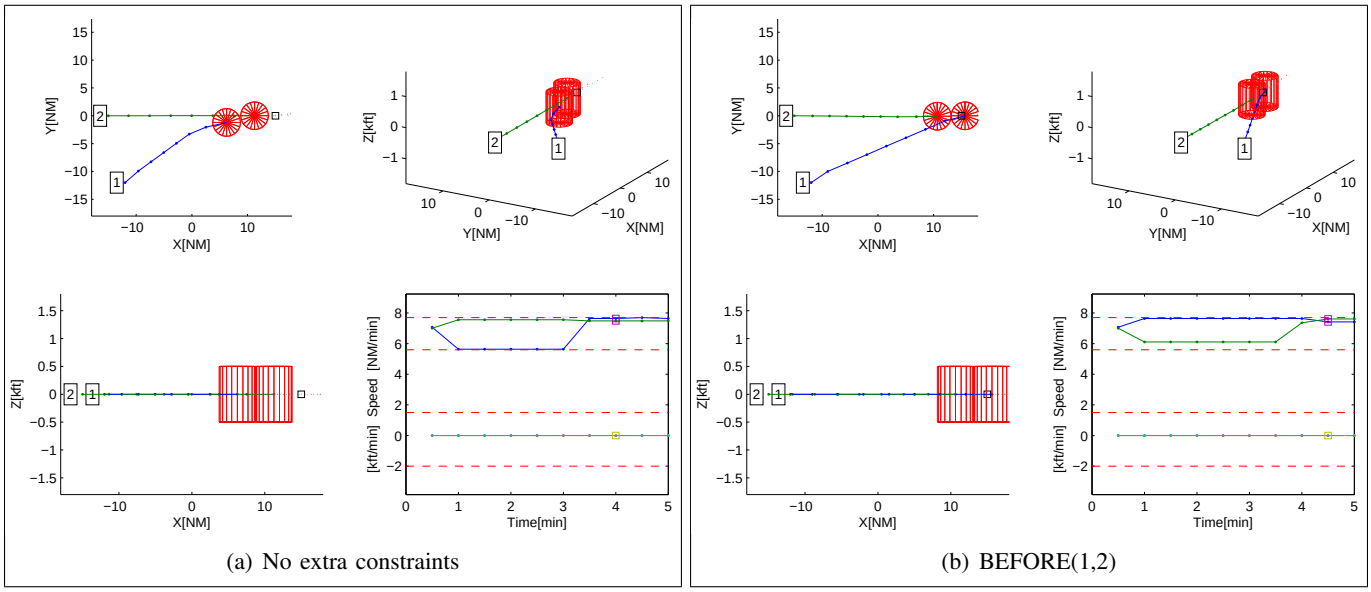
(c) With OVER(1,2) tactic

(d) HORIZSEP(1,2) tactic

(e) HOIRZSEP(1,2) and BEFORE(1,2) tactics

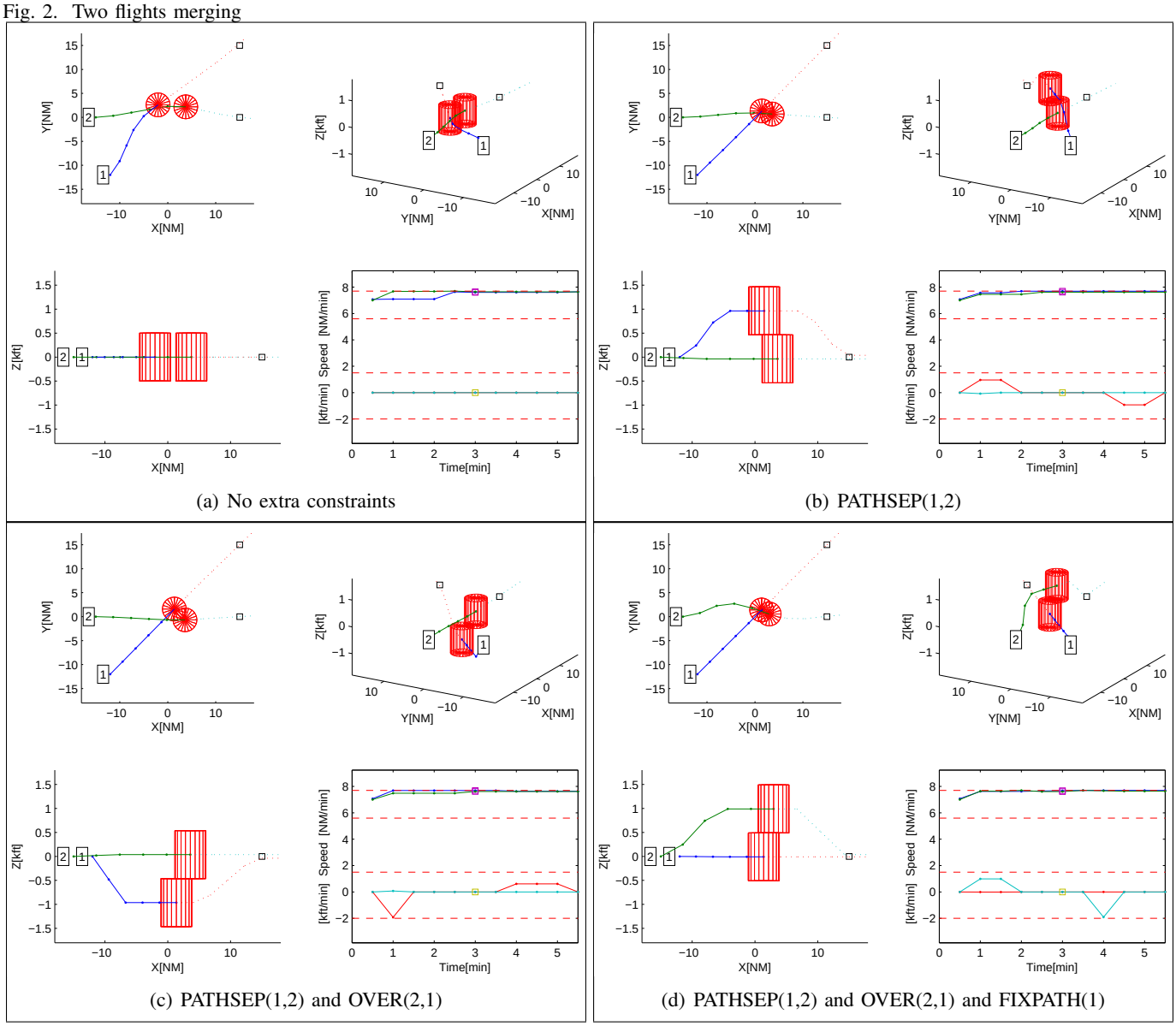
(f) FIXPATH tactics for both

Fig. 1. Flight 1 climbing and crossing Flight 2



(a) No extra constraints (b) BEFORE(1,2)

Fig. 2. Two flights merging



(a) No extra constraints (b) PATHSEP(1,2)

(c) PATHSEP(1,2) and OVER(2,1) (d) PATHSEP(1,2) and OVER(2,1) and FIXPATH(1)

Fig. 3. Crossing at same level

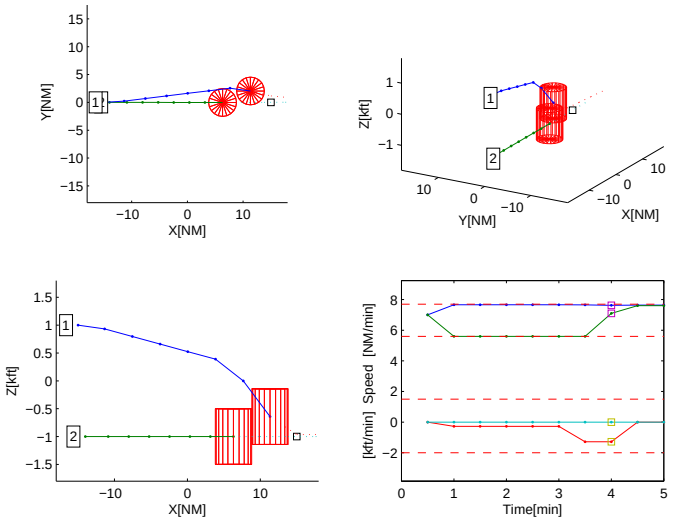


Fig. 4. Descending into merge, with BEFORE(1,2) and FIXPATH(2)

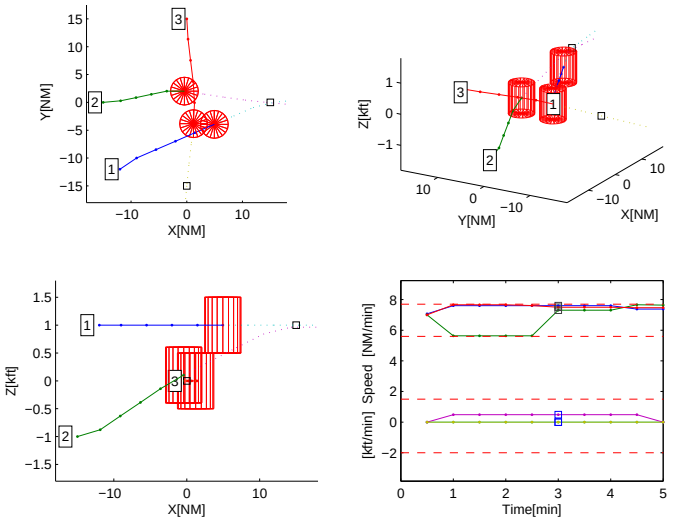


Fig. 5. Three flights example, with HORIZSEP(2,3) constraints

same altitude merge on to a common target, typical on arrival routes into airports. With the paths constrained to intersect at the merge point, spatial resolution is not applicable here, so only temporal is considered. With no supervisory constraints applied, the optimizer returns the solution seen in Fig. 2(a), with aircraft 2 going ahead while 1 slows to merge behind. If the supervisor desires the opposite order, the BEFORE(1,2) constraint can be applied, yielding the solution in Figure 2(b), which does indeed show the opposite order after the merge with 1 speeding up and 2 slowing down to achieve it.

Figure 3 shows a scenario involving two aircraft crossing at the same altitude. With no supervision, aircraft 1 crosses behind aircraft 2, both remaining at the same altitude throughout, as seen in Figure 3(a). If spatial separation is desired, the PATHSEP constraint provides the result seen in Figure 3(b). Since the two paths cross horizontally, the only way to achieve spatial separation is to change the altitudes, and in this case aircraft 1 has climbed up and over aircraft 2. If the opposite order is desired, the OVER(2,1) can be added, resulting in the solution shown in Figure 3(c). Finally, seeking to prevent aircraft 1 from changing altitude at all, the FIXPATH(1) constraint is added, and Figure 3(d) shows the result in which aircraft 2 has to climb over the path of aircraft 1.

Figure 4 shows a scenario in which the merge is done vertically, with aircraft 1 descending to merge with aircraft 2. In this example, a FIXPATH(2) constraint has also been included to prevent aircraft 2 using a dog-leg manoeuvre to give way to aircraft 1.

Finally, to investigate scalability, Figures 5 and 6 show examples with three and four aircraft, respectively. These cases were inspired by situations just west of Bristol, UK, where flights from the northern UK to Spain cross

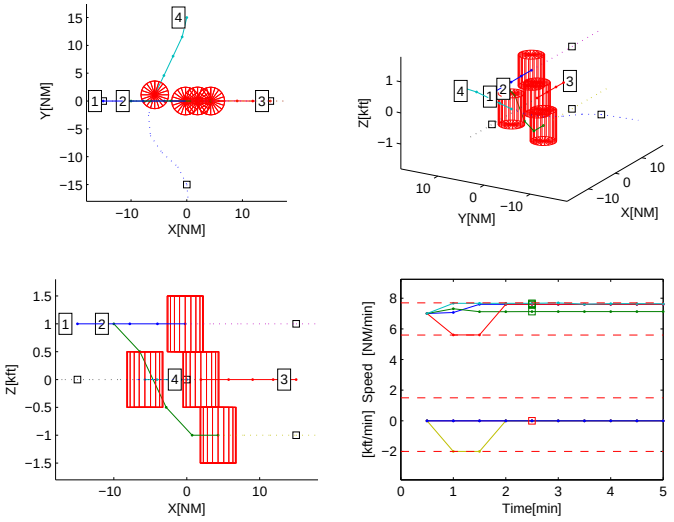


Fig. 6. Four flights example, with constraints FIXPATH(3), OVER(3,2) and BEFORE(4,3)

transatlantic routes in and out of London Heathrow and other European hubs. Some example constraints have been applied and all have been followed correctly. These examples illustrate the ability of the optimization to handle global conflict resolution, *i.e.* beyond pairwise [5].

Table III shows solution times for the examples shown and some others. Timing was measured simply using the elapsed wall time to run the solver, and will include overheads for reading and writing data and processing the AMPL model. A time limit setting of 60s was applied to the Gurobi solver, after which the solver returned the best feasible solution available. Although not a thorough investigation of complexity, some patterns can be seen. In particular, the PATHSEP and BEFORE constraints cause significant extra computing, likely to be linked to the many additional binary variables d_t they introduce,

for all pairs of time steps. FIXPATH also introduces additional binaries d_p, but the impact on computation is less severe. Observe that the binaries d_p in the FIXPATH case do not appear multiplied by the "big-M" factor, hence FIXPATH can offer tight relaxations of the MILP problem and fast solution time. Although many of the solution times are very fast, others are far slower, and the inherent unpredictability of solution time means the method is not suitable for immediate practical use. However, algorithmic enhancements [26] and the on-going development of parallel and distributed solvers [27] indicate potential for practical computation in the future.

TABLE III
COMPUTATION TIMES

Constraints	Sol. time (s)
Fig. 1	
IGNORE(1,2)	0.34
No tactics	1.15
OVER(1,2)	0.60
HORIZSEP(1,2)	0.69
HORIZSEP(1,2); BEFORE(1,2)	15.56
FIXPATH(1); FIXPATH(2)	1.68
Fig. 2	
No extra constraints	0.95
BEFORE(1,2)	3.77
Fig. 3	
No tactics	0.69
PATHSEP(1,2)	60.33
PATHSEP(1,2); OVER(2,1)	60.37
PATHSEP(1,2); OVER(2,1); FIXPATH(1)	60.36
Fig. 4	
No tactics	1.15
BEFORE(1,2)	8.68
BEFORE(1,2); FIXPATH(2)	21.23
Fig. 5	
No tactics	8.36
HORIZSEP(2,3)	9.24
FIXPATH(2); FIXPATH(3)	4.44
Fig. 6	
No tactics	5.73
FIXPATH(3)	12.52
FIXPATH(3); OVER(3,2)	4.90
FIXPATH(3); OVER(3,2); BEFORE(4,3)	29.66

V. IMPLEMENTATION DISCUSSION

The illustrations in Figs. 1 to 6 are intended to illustrate the workings of the constraints and do not provide a full design for a potential human-machine interface (HMI). It is anticipated that the trajectory optimizer could be implemented in a Controller Working Position (CWP) as part of an enhanced Medium Term Conflict Detection (MTCD) [28] system, either within the display tools or as a server to an CWP client. Application of a constraint would be achieved by selecting an aircraft on the plan display and the appropriate type of constraint, followed by a second aircraft if necessary. For example, to apply BEFORE(1,2), first select aircraft 1, then click "BEFORE", then select aircraft 2. In the context of a separation monitor display [29] in which a pair of aircraft are represented by a single symbol, application of a pairwise constraint would be done with two clicks: the pair and the constraint type. Since the resulting trajectory is likely to involve multiple manoeuvres, commands to the aircraft involved are best suited to a future environment using 4D trajectory datalink [3]. The optimizer does not use fixed waypoints so the method is suited to both free route or structured airspace. For the latter, it would be interesting to investigate further constraints exploiting waypoints when available, to reduce controller workload. However this is beyond the scope of this paper.

VI. CONCLUSIONS

This paper has proposed constraints to allow a human air traffic controller to interact with a trajectory optimizer. The constraints allow the specification of the relative sense (ahead/behind/over/under) of conflict resolution between aircraft, or the class of the resolution (vertical/horizontal/temporal). Inspired by the playbook approach to automation, the method offers a way of sharing authority between the controller and the optimizer, capturing expert knowledge of the controller without losing the flexibility of the optimizer.

REFERENCES

[1] SESAR Consortium, "The ATM Target Concept - D3," SESAR Consortium, Tech. Rep., 2007.

[2] V. Cox and M. Romanowski, *NextGen Implementation Plan*. Federal Aviation Administration, 2009.

[3] SESAR Consortium, "SESAR concept of operations," SESAR, Tech. Rep. DLT-0612-222-01-00, 2007.

[4] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE transactions on systems, man, and cybernetics. Part A, Systems and humans : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 30, no. 3, pp. 286–97, May 2000.

[5] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, vol. 1, no. 4, pp. 179–189, December 2000.

[6] A. Alonso-Ayuso, L. F. Escudero, and F. J. Martín-Campo, "Collision avoidance in air traffic management: A mixed-integer linear optimization approach," *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –11, 2011.

[7] A. Alonso-Ayuso, L. Escudero, and F. Martin-Campo, "On modeling the air traffic control coordination in the collision avoidance problem by mixed integer linear optimization," *Annals of Operations Research*, pp. 1–17, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10479-013-1347-y

[8] L. Pallottino, E. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 3, no. 1, pp. 3 –11, mar 2002.

[9] M. Christodoulou and S. Kodaxakis, "Automatic commercial aircraft-collision avoidance in free flight: The three-dimensional problem," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 2, pp. 242–249, 2006.

[10] J. Omer and T. Chaboud, "Automated conflict-free planning: Experiments on real air traffic data," in *28th International Congress of the Aeronautical Sciences*, Brisbane, Australia, 2012.

[11] S. E. Campbell, "Multiscale path optimization for the reduced environmental impact of air transportation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 3, pp. 1327 –1337, sept. 2012.

[12] X.-B. Hu, S.-F. Wu, and J. Jiang, "On-line free-flight path optimization based on improved genetic algorithms," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 8, pp. 897 – 907, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197604001009

[13] A. G. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proceedings of American Control Conference*, Anchorage, Alaska, 2002, pp. 1936–1941.

[14] A. E. Vela, S. Solak, J.-P. B. Clarke, W. E. Singhose, E. R. Barnes, and E. L. Johnson, "Near real-time fuel-optimal en route conflict resolution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 826–837, 2010.

[15] W. Hylkema and H. Visser, "Aircraft conflict resolution taking into account controller workload using mixed integer linear programming," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003.

[16] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[17] C. Miller, H. Funk, P. Wu, J. Meisner, and M. Chapman, "The playbook approach to adaptive automation," in *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 49, no. 1. Human Factors and Ergonomics Society, 2005, pp. 15–19.

[18] C. A. Miller and R. Parasuraman, "Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 49, no. 1, pp. 57–75, 2007.

[19] Gurobi Optimization Inc., "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com

[20] A. Richards and O. Turnbull, "Inter-sample avoidance in trajectory optimizers using mixed-integer linear programming," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 4, pp. 521–526, 2015.

[21] A. Nuic, D. Poles, and V. Mouillet, "BADA: An advanced aircraft performance model for present and future ATM systems," *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 10, pp. 850–866, 2010.

[22] O. Turnbull and A. Richards, "Examples of supervisory interaction with route optimizers," in *Second SESAR Innovation Days*, Braunschweig, Germany, November 2012.

[23] A. G. Richards and J. P. How, "Robust variable horizon model predictive control for vehicle maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 16, pp. 333–351, 2006.

[24] G. Ferrari-Trecate, P. Letizia, and M. Spedicato, "Optimization with piecewise-affine cost functions," ETH, Tech. Rep. AUT01-13, June 2001. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.715.20&rep=rep1&type=pdf

[25] O. Turnbull and A. Richards, "4-d trajectory optimizers for conflict avoidance using speed advisories," in *Tenth USA/Europe Seminar on ATM R&D*, Chicago, June 2013. [Online]. Available: http://www.atmseminar.org/papers.cfm?seminar_ID=10

[26] M. Earl and R. D'Andrea, "Iterative MILP methods for vehicle-control problems," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1158–1167, 2005.

[27] R. Carvajal, S. Ahmed, G. Nemhauser, K. Furman, V. Goel, and Y. Shao, "Using diversification, communication and parallelism to solve mixed-integer linear programs," *Operations Research Letters*, vol. 42, no. 2, pp. 186 – 189, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167637714000236

[28] EUROCONTROL, "Specification for medium-term conflict detection," EUROCONTROL, Tech. Rep. EUROCONTROL-SPEC-0139, 2010. [Online]. Available: http://www.eurocontrol.int/publications/medium-term-conflict-detection-mtcd-specification

[29] A. Roberts and S. Pember, "Air traffic control system," US Patent 9,245,451, January, 2016. [Online]. Available: https://www.google.com/patents/US9245451

AUTHOR BIOGRAPHIES

Oliver D. N. Turnbull PhD. Engineering Mathematics, University of Bristol, 2008; MEng Avionic Systems Engineering, University of Bristol, 2004. After completing his PhD he worked as a space systems engineer before returning to the University of Bristol as a Research Assistant with interests include optimization for vehicle guidance, path planning and decision making in the presence of uncertainty. He is now a Mission Analyst with Deimos Space.

Arthur G. Richards received the M.Eng. degree from Cambridge University in 2000 and the SM and PhD degrees from MIT in 2002 and 2004, respectively. Since 2004, he has been with the Department of Aerospace Engineering, University of Bristol, Bristol, U.K., where he is currently a Reader. His research interests include trajectory optimization, model predictive control, robotics, and their combination to develop high-performance guidance for autonomous vehicles.